DISCRIMINANT ANALYSIS AND SUPERVISED VECTOR QUANTIZATION FOR CONTINUOUS SPEECH RECOGNITION

George Yu, William Russell, Richard Schwartz, John Makhoul

BBN Systems and Technologies Corp.
Cambridge MA 02138

ABSTRACT

In this paper we describe several attempts to improve the recognition accuracy with the use of supervised clustering techniques. These techniques modify the distance metric and/or the clustering procedure in a discrete HMM recognition system in an attempt to improve phonetic modeling. We considered three techniques: Linear Discriminant Analysis, a hierarchical supervised vector quantization technique, and Kohonen's LVQ2 technique. All experiments were performed on the DARPA resource Management Corpus using the BBN BYBLOS system [5]. Even though the techniques improved the phonetic recognition capability of the vector quantization, the overall word and sentence recognition accuracy did not improve.

I INTRODUCTION

Even in a discrete Hidden Markov Model system, there is an underlying distance metric that is used to divide the spectral space into distinct regions. The BBN BYBLOS Continuous Speech Recognition System currently uses context-dependent phonetic discrete HMMs based on three codebooks. The first codebook contains 14 mel-frequency warped cepstral coefficients (c1-c14) computed every 10 ms directly from the speech power spectrum. The second codebook contains the 14 "differences" of these parameters, derived by computing the slope of a least squares linear fit to a five-frame window centered around each frame [1]. Finally, we use a third codebook that has the amplitude-normalized log rms energy and the "difference" of this energy. We divide the 30 features among three codebooks to avoid the training problem associated with high dimensionality. Each codebook is designed using a nonuniform binary clustering algorithm, followed by several iterations of the k-means algorithm [2]. The k-means clustering algorithm implicitly uses Euclidean distance.

It has been suggested that it is possible to improve recognition accuracy by performing linear discriminant analysis [3, 4]. We also consider several methods of nonlinearly warping the spectral space as part of the vector quantization process. We call these methods "supervised clustering" techniques. To use these techniques, we need to define the classes that we want to discriminate. We chose the (50 or so) basic phonemes as that set, under the assumption that these model most of the distinctions that must be made in large vocabulary speech recognition. To obtain phoneme labels for the training data we first estimate speech models using the standard techniques in the BYBLOS system and then segment all of the training data into phonemes automatically using the decoder (recognizer) constrained to find the correct answer. The recognized segment boundaries are then used to assign a phoneme label to each frame. Each of the techniques described below then attempts to define a distance metric or vector quantizer that can recognize the phoneme label of a single frame of speech.

In Chapter 2 we describe how we use linear discriminant analysis to modify the spectral parameters before vector quantization. In Chapter 3, we describe two methods for nonlinear supervised clustering. All experiments and results are presented in Chapter 4.

II LINEAR DISCRIMINANT ANALYSIS

Brown [3] has proposed using several successive frames jointly in order to model the joint density of the observed speech more accurately. He then uses linear discriminant analysis (LDA) to reduce the number of dimensions. We attempted to use LDA on our 30 mixed features to find a set of features that would, in fact be more independent. In addition, we might hope that we would automatically find a more beneficial weighting of the different features than simple Euclidean distance.

We compute the within (phoneme) class and between class means and covariances of all the frames of training data. We use the generalized eigenvector solution to find the best set of linear discriminant features. Then, we simply cluster and quantize the 30 new features as usual. Alternatively, we can divide the new features up into a small number of codebooks in order to reduce the quantization error. We use these new (quantized) features in place of the original features for discrete HMM continuous speech recognition.

III SUPERVISED VECTOR QUANTIZATION

In addition to simple linear discriminants, we consider more complex warpings of the feature space. We call the general approach *supervised clustering* or *supervised VQ*. Instead of finding a codebook that minimizes mean square error, without regard to phonetic similarity, we use the training data to generate a codebook that tends to preserve differences that are phonetically important, and disregard feature differences (even if they are large) that are not phonetically important. In effect,

CH2847-2/90/0000-0685 $1.00 © 1990 IEEE

we attempt to maximize the mutual information between the VQ clusters and phonetic labels. We describe two techniques below that seem suitable for accomplishing this goal.

III.1 Binary Division of Space

The first algorithm is most closely related to the nonuniform binary clustering algorithm that we use to derive an initial estimate for k-means clustering [2]. All the labeled frames are initially placed in one cluster. Then, we iteratively divide the clusters until we have the desired number. One of the many clustering algorithms we tried is given below.

First we measure the entropy reduction that would result from dividing a single cluster into two:

1. Estimate a single diagonal-covariance Gaussian for the frames with each phoneme label in the cluster.
2. Identify the two most "prominent" phonemes within the cluster.
3. Divide all the frames in the cluster into two new clusters using these two guassian distributions.
4. Compute the difference between the entropy of the phoneme labels in the original cluster, and the average entropy of the two new clusters, weighted by the number of samples in each subcluster.

The outer loop repeatedly divides the cluster that will result in the largest enropy reduction.

1. Divide the cluster that would result in the largest entropy reduction.
2. Create two new clusters and measure the potential entropy reduction for dividing each of the two resulting clusters as described above.
3. If we have fewer than 256 clusters, go to (1)

The one-step lookahead avoids dividing a large cluster when no reduction in entropy would result. The resulting codebook is then used to quantize all of the training and test data. While this algorithm increased the mutual information between the codebook and the phonetic labels, there was no gain in the overall recognition accuracy.

III.2 LVQ2: Kohonen's Learning Vector Quantizer

The LVQ2 algorithm [6] was used very effectively in a phoneme recognition system [7]. The algorithm amounts to a discriminative training of the codebook's means to maximize recognition of the frame labels.

As before, we start with the set of phonetically labeled frames. For LVQ2, we use a "sliding window" of some fixed size centered around each frame to create large feature vectors. (7-frame windows were used in [7].) Then we use the binary and k-means algorithm to divide the feature vectors from each phoneme into several clusters. We make the number of clusters for each phoneme proportional to the square root of the number of frames of that phoneme, such that the total number of clusters is 256. Each cluster has the name of the phoneme data in it. Then, we use LVQ2 to shift the cluster means to optimize frame recognition. We review the algorithm below briefly. For each feature vector:

1. Find the nearest cluster and the next nearest cluster from a different phoneme.
2. If the nearest cluster is from the wrong phoneme and the second nearest is of the correct phoneme, shift the mean of the correct cluster toward the feature vector in question and shift the wrong cluster's mean away, according to:

$$\mathbf{m}_i(t + 1) = \mathbf{m}_i(t) - \alpha(t)\ (\mathbf{x}(t) - \mathbf{m}_i(t))$$

$$\mathbf{m}_j(t + 1) = \mathbf{m}_j(t) + \alpha(t)\ (\mathbf{x}(t) - \mathbf{m}_j(t))$$

where $\mathbf{x}$ is a training vector belonging to class j,
$\mathbf{m}_i$ is the reference vector for the incorrect category,
$\mathbf{m}_j$ is the reference vector for the correct category, and
$\alpha(t)$ is a monotonically decreasing function of time.

The above algorithm is iterated until convergence (which requires some care). As suggested in the reference, we used several adjacent speech frames together as a longer feature vector. We also performed experiments in which we used the LVQ2 algorithm separately on several frames of steady state cepstra and several frames of difference cepstra. The resulting codebooks are used in the normal way.

IV EXPERIMENTS AND RESULTS

We performed speaker-dependent recognition experiments on a six-speaker subset of the DARPA Resource Management speech corpus, using the May, 1988 test set. Continuous speech recognition experiments were done using the word-pair grammar and also with no grammar.

To understand the behavior of the supervised clustering algorithms, we measure the correspondence between the resulting codebooks and the phonetic labels. First, we determine the most frequently occurring phoneme label within each cluster. We then quantize new frames and use the VQ indices to "recognize" the phoneme labels of the independent data. Table 1 shows the phoneme-frame recognition accuracy for training and test data, and for steady state and difference cepstra. The results show that codebooks made by the binary split algorithm are slightly better than the unsupervised k-means algorithm at recognizing a phoneme label from a single frame of steady-state cepstra. We performed similar frame recognition experiments using LVQ2 with sliding windows of length 1, 3, 5, and 7 frames. Table 1 shows that LVQ2 is better than both K-means and binary division at predicting frame phoneme labels, even when LVQ2 does not look at a frame's neighbors. As we increase the window size, the correspondence between the VQ clusters and phonemes improves significantly.

Metric / Algorithm		Cepstra		Diff Cepstra	
		train	test	train	test
K-means	(1 frm)	43%	41 %	23 %	21 %
Binary	(1 frm)	45	42	25	22
LVQ2	1 frm	49	44	29	25
LVQ2	3 frm	57	51	39	33
LVQ2	5 frm	62	55	46	39
LVQ2	7 frm	65	57	50	43

Table 1. Frame Phoneme Recognition Rate

The results of continuous speech recognition experiments are given in Table 2. The control experiment for these results which used a somewhat older version of the BYBLOS system is labeled k-means. The HMM recognition system has a number of system parameters. Wherever possible, these parameters are left unchanged between the k-means control and the other tests.

The last condition in Table 2 (labeled "Recent BBN") corresponds to the most recently reported performance of the BBN BYBLOS system for this subset of the May, 1988 data [8]. The system that produced this performance is similar to the control system, except that in addition to modeling coarticulation within words we used cross-word context-dependent phonetic models (triphones) to model coarticulation between words. This experimental result is included purely for reference.

We discuss four LDA experiments with variations in the number of codebooks and assignment of linear discriminants to codebooks. In all four tests we concatenated the 14 cepstral coefficients, the 14 "difference" coefficients, and the two normalized energy coefficients, and used LDA to extract a new set of 30 discriminant features. In the first test (30f) we then clustered all 30 discriminant features into one codebook which was used in HMM recognition. In the second test (15f,15f), we split the 30 discriminant features into two 15-parameter codebooks. In the third test (15f) we used only the first 15 discriminant features in a single codebook. Finally, in the fourth test (km,15f), we used the standard three codebooks together with a fourth codebook containing the first 15 discriminant features. As can be seen in Table 2, most results using LDA did not improve significantly over the baseline 3-codebook condition.

System		Grammar	
	Codebk(s)	word-pair	none
K-means (km)	c,d,e	3.6 %	18.8 %
Lin Discrim	30f	5.1	20.3
Lin Discrim	15f,15f	4.7	20.9
Lin Discrim	15f	6.2	24.9
Lin Discrim	km,15f	3.8	16.1
Binary Div	30f	4.8	20.6
Binary Div	c,d,e	3.9	17.2
Binary Div	km,30f	3.3	17.1
LVQ2 (1 frm)	c,d,e	4.0	18.5
LVQ2 (3 frm)	c,d,e	4.2	17.7
LVQ2 (5 frm)	c,d,e	3.3	18.1
LVQ2 (7 frm)	c,d,e	3.9	18.6
LVQ2 (3/5 frm)	c,d,e	3.2	18.1
Recent BBN	c,d,e	2.1	13.7
c = cep, d = dif, e = energy, <#>f = <#> features			

Table 2. Word Recognition Error: Multiple codebooks.

We performed three recognition tests of supervised clustering using the binary division algorithm. In the first test (30f), we concatenated all 30 parameters into a single feature vector and created one codebook using binary division. In the second test (c,d,e), we used binary division to cluster separately the cepstrum, difference cepstrum, and energy coefficients. This three-codebook experiment, then, is a direct comparison between binary division and unsupervised K-means. As Table 2 shows, neither the one-codebook nor the three-codebook binary division experiments resulted in improved recognition over the baseline. As a third test (km,30f), we add the 30-feature binary-division codebook from the first test to the three unsupervised codebooks of the control. This four codebook experiment results in a small (10%) reduction in error rate relative to the three codebooks by themselves.

Next, we show several three-codebook recognition experiments using the LVQ2 algorithm with windows of size 1, 3, 5, and 7 frames on the cepstral and difference coefficients. For simplicity, the energy parameters were clustered using the baseline K-means algorithm. In the last experiment, the cepstra codebook uses a 3 frame window and the difference-cepstra codebook uses a 5 frame window. While there are small random variations in the results, there are no significant improvements in overall recognition accuracy.

We were surprised that the LVQ2 algorithm improved the frame recognition accuracy so much without improving the overall speech recognition accuracy. Therefore, we performed an additional set of experiments using only one codebook with steady-state cepstra. These results are summarized in Table 3. We see, again, that the improvement in cluster/phoneme correspondence from increasing LVQ2's window size does not necessarily translate into better system word recognition.

System (cepstra–only)		Grammar word-pair	none	frame recog
K-means	(1 frm)	8.4 %	30.0 %	41 %
LVQ2	3 frm	7.5	30.8	51
LVQ2	5 frm	7.5	28.9	55
LVQ2	7 frm	8.2	29.8	57

Table 3. Word Recognition Error vs Frame Recognition Accuracy for one codebook.

V DISCUSSION OF RESULTS

The results generally show that, even when the supervised clustering is successful at improving the correspondence between the VQ codebook regions and the phonetic labels, the overall speech recognition accuracy does not improve. We can draw two possible conclusions from these results relative to previous successes with these techniques. First, while it might be possible to find a small number of discriminant directions that are important for a small vocabulary task - especially one with minimal pair differences - it may not be as easy in a large vocabulary task, where the important distinctions are many and also very varied. That is, any choice of discriminants that is better for some distinctions may be worse for others. Second, it is not clear that optimizing phonetic distinctions on single frames will help a recognition system whose goal is to recognize words using triphone models. Katagiri has reported [9] that subsequent attempts to use LVQ2 to improve the phonetic models for digit recognition were unsuccessful. The improvements returned when LVQ2 was used to optimize the digit recognition directly.

VI CONCLUSIONS

We have experimented with several techniques for supervised clustering, based on improving the phonetic modeling capability of the codebooks. While the codebooks themselves were improved, the overall recognition accuracy did not improve. In the future, we will attempt to use similar techniques within the context of the whole training and recognition system.

Acknowledgement

The authors would like to thank Robin Rohlicek of BBN for several useful discussions. This work was supported by the Defense Advanced Research Projects Agency and monitored by the Office of Naval Research under Contract Nos. N00014-85-C-0279 and N00014-89-C-0008.

REFERENCES

[1] S. Furui (1986) "Speaker-Independent Isolated Word Recognition Based on Emphasized Spectral Dynamics," *IEEE ICASSP-86, pp. 1991-1994*

[2] Makhoul, J., S. Roucos, and H. Gish (1985) "Vector Quantization in Speech Coding" *Proc. of IEEE, Vol. 73, No.11, pp.1551-1588*

[3] Brown, P. (1987) "The Acoustic-Modeling Problem in Automatic Speech Recognition" *PhD Thesis, CMU, 1987*

[4] G. Doddington (1989) "Phonetically Sensitive Discriminants for Improved Speech Recognition," *IEEE ICASSP-89, pp. 556-559*

[5] Chow, Y., M. Dunham, O Kimball, M. Krasner, G.F. Kubala, J. Makhoul, P. Price, S. Roucos, and R. Schwartz (1987) "BYBLOS: The BBN Continuous Speech Recognition System," *IEEE ICASSP-87, pp. 89-92*

[6] Kohonen, T., G. Barna, and R. Chrisley (1988) "Statistical Pattern Recognition with Nerual Networks: Benchmarking Studies," *IEEE, Proc. of ICNN, Vol. I, pp. 61-68, July, 1988*

[7] McDermott, E. and S. Katagiri (1989) "Shift-Invariant, Multi-Category Phoneme Recognition using Kohonen's LVQ2," *IEEE ICASSP-89, pp. 81-84*

[8] S. Austin, C. Barry, Y-L. Chow, A. Derr, O. Kimball, F. Kubala, J. Makhoul, P. Placeway, W. Russell, R. Schwartz, and G. Yu (1989) "Improved HMM Models for High Performance Speech Recognition," *Proceedings of the DARPA Speech and Natural Language Workshop, October, 1989*

[9] S. Katagiri, Personal Communication (October, 1989)